OPTIMIZING CLOUD STORAGE WITH THE UTILIZATION OF DYNAMIC DEDUPLICATION TECHNIQUES

Madhu Ramteke[1], K.L. Sinha[2]

[1,2]Computer Science & Engineering, CSVTU (India)

ABSTRACT

Cloud computing performs a maximum essential job within the industry area in nowadays as computing resources are introduced as a utility on demand to customers over the internet. Cloud storage is probable one of the offerings supplied in cloud computing which has been increasing in reputation. The important advantage of utilizing cloud storage from the clients' factor of view is that customers can decrease their expenditure in shopping and preserving storage infrastructure on the equal time as paying the quantity of storage asked, which may be scaled-up and down upon demand. With the growing records length of cloud computing, a discount in data volumes ought to assist vendors to reduce the expenses of walking big storage system and saving strength intake. So information deduplication systems had been added to beautify storage performance in cloud storages. With the dynamic nature of facts in cloud storage, statistics usage in cloud alterations time beyond regulation, some information chunks could also be read in all likelihood in time frame, but will now not be used in but yet again length. Some datasets can be usually retrieved or up to date thru various customers whilst, at the same time others may also want the excessive degree of redundancy for reliability requirement. For that reason, it is essential to aid this dynamic function in cloud storage. However modern techniques are usually targeted on dynamic scheme with much less parameter. To better ensure information optimization, the proposed system makes the number one try and officially copes with the problem of storage space by adding more parameter inside the present device; we take into account the converting of consumer's demand of documents. An element in redundancy manager will monitor file access activities.

Index Terms: Deduplication, Cloud Computing, Cloud Storage, Availability.

I. INTRODUCTION

Cloud computing, where application and files are hosted on a "cloud" consisting of thousands of computers and servers, all linked together and accessible via the internet. The idea of cloud computing is quite similar to grid computing, which pursuits to reap useful resource virtualization [1]. In grid computing, the companies sharing their computing resources, inclusive of processors, in an effort to achieve the most computing potential, while cloud computing pursuits to offer computing assets as a utility on demand, that can scale up or down at any time, to more than one customers. This makes cloud computing play a chief role in the enterprise domain, while grid is famous in instructional, scientific and engineering studies [2].

Many definitions of cloud computing have been described, depended on the man or woman factor of view or generation used for system improvement. In widespread, we are able to outline cloud computing as a enterprise version that offer computing sources as a service on demand to customers over the Internet [3].

The important characteristics of cloud computing have been defined in [3]. Cloud vendors pool computing assets collectively to serve clients via a multi-tenant version. Computing resources are introduced over the Internet in which clients can access them through numerous customer structures. Customers can get entry to the sources on-demand at any time without human interplay with the cloud provider. From a customers' point of view, computing sources are countless, and client demands can hastily trade to meet business targets. This is facilitated by way of the potential for cloud offerings to scale resources up and down on call for leveraging the strength of virtualization. Moreover, cloud companies are able to screen and manipulate the use of sources for each customer for billing functions, optimization resources, capability planning and other tasks. Cloud storage is one of the offerings in cloud computing which affords virtualized storage on demand to customers.

Cloud storage may be used in lots of one of a kind ways [4]. For example, customers can use cloud storage as a backup carrier, rather than keeping their very own storage disks. Organizations can pass their archival storage to the cloud which they can obtain greater capacity at the low-price, in place of shopping for additional bodily storage. Applications walking within the cloud also require transient or everlasting records storage in an effort to guide the programs.

As the amount of information inside the cloud is unexpectedly increasing, clients assume to reach the on-demand cloud offerings at any time, whilst carriers are required to hold system availability and method a massive amount of data. Providers want a manner to dramatically reduce data volumes, so we can reduce prices whilst saving energy intake for going for walks huge storage structures. Similar to different storages, storage in cloud environments also can use records deduplication technique.

Data deduplication is a technique whose objective is to enhance storage efficiency. With the goal to lessen storage space, in traditional deduplication structures, duplicated facts chunks discover and shop handiest one reproduction of the statistics in storage. Logical pointers are created for different copies in place of

storing redundant data. Deduplication can reduce both storage area and community bandwidth [7]. However such techniques can end result with a poor impact on machine fault tolerance.

Because there are numerous files that consult with the equal facts chunk, if it turns into unavailable due to failure can result in reduced reliability. Due to this problem, many processes and strategies were proposed that no longer handiest provide solutions to achieve storage efficiency, however additionally to improve its fault tolerance. These strategies offer redundancy of data chunks after performing deduplication. However, modern data deduplication mechanisms in cloud storage are static schemes implemented agnostically to all facts situations. This is a trouble as records situations exhibit extraordinary statistics traits that require exceptional levels of fault tolerance necessities.

For instance, records utilization in cloud changes additional time; some facts chunks may be study frequently in a time period, however won't be utilized in some other period. Due to the downside of static schemes, which cannot address converting person behavior, deduplication in cloud storages calls for a dynamic scheme which has the ability to conform to diverse get right of entry to styles and changing person behavior in cloud storages. The contribution of this paper is a dynamic records deduplication scheme for cloud storage, as a way to fulfil a stability among storage performance and fault tolerance requirements, and also to enhance performance in cloud storage systems that experience modifications in records situations and person patterns. The relaxation of this paper is organized as follows: segment II provides heritage principles and associated work. Section III demonstrates a proposed device version. Section IV illustrates the simulation of the proposed system version. Section V describes the experimental end result. Section VI discusses the future work. Finally phase VII concludes this paper.

II. BACKGROUND AND RELATED WORK

A. Deduplication in Cloud Storages

Data deduplication is a technique to reduce storage area. By figuring out redundant statistics the use of hash values to compare records chunks, storing handiest one replica, and creating logical hints to other copies as opposed to storing different actual copies of the redundant statistics [5], [6]. Deduplication reduces records quantity so disk space and network bandwidth may be reduced which reduce prices and power consumption for walking storage systems [7]. Data deduplication may be implemented at nearly every factor which records is saved or transmitted in cloud storage [7]. Many cloud carriers offer catastrophe recuperation [8] and deduplication can be used to make disaster recovery extra effective by using replicating information after deduplication for speeding up replication time and bandwidth value financial savings. Backup and archival storage in clouds can also follow data deduplication to be able to lessen physical ability and network visitors [9], [10]. Moreover, in stay migration method, we need to transfer a large quantity of duplicated reminiscence photograph facts [11]. There are 3 major performance metrics of migration to recall: overall statistics transferred, overall migration time and carrier downtime. Longer migration time and downtime could be lead to provider failure. Thus, deduplication can help in migration [12]. Deduplication may be used to reduce storage of lively records such as digital system snap shots. Factors to recall when the usage of deduplication in primary storage is a way to stability the alternate-offs between storage area saving and overall performance effect [13].

Additionally, Mandagere, et al., [13] nation that deduplication algorithms reflect the performance of deduplicated storage in terms of fold thing, reconstruction bandwidth, metadata overhead, and useful resource usage.

B. Dependability Issues When acting deduplication, a part of facts chunks are a lot greater essential than others (For example, information chunks that are referenced through many documents). Traditional deduplication tactics do no longer put into effect redundancy of facts chunks. Thus, deduplication may additionally reduce the reliability of the storage device due to the loss of a few vital chunks that can cause the loss of many documents. As a result, the essential chunks should be replicated more than the much less vital information chunks with a purpose to improve reliability of the system. The authors in [14], remember the outcomes of deduplication at the reliability of the archival device. They proposed an method to improve reliability by using growing a method to weigh and degree the significance of each chew through inspecting the variety of information files that percentage the chew, and use this weight to pick out the level of redundancy required for the bite to guarantee QoS.

C. Related Work

Looking at system architectures of current works of deduplication for cloud backup offerings including SAM [10],AA-Dedupe [15], CABdedupe [16], and SHHC [17].

SAM [10] device structure consists of 3 subsystems: File Agent, Master Server and Storage Server. Clients join backup offerings, then File Agents are distribute and set up on their machines, even as provider provider gives Master Server and Storage Server in datacentre to serve the backup requests from clients. Most of present solutions that use deduplication technology in most cases cognizance on the reduction of backup time while ignoring the healing time. The authors proposed CABdedupe [16], a performance booster for both cloud backup and cloud restore operations, which is a middleware that is orthogonal and can be incorporated into any present backup system. CABdedupe includes CAB-Client and CAB-Server, that's positioned on the authentic consumer and server modules in present backup systems. The primary aim of these related works are the subsequent: SAM aims to acquire an foremost change-off among deduplication performance and deduplication overhead, CABdedupe reduces both backup time and recovery time.

AA-Dedupe [15] goals to lessen the computational overhead, growth throughput and switch performance, at the same time as SHHC [17] tries to improve fingerprint storage and research mechanism, however has a issue of scalability. SHHC is a novel Scalable Hybrid Hash Cluster designed for enhancing response times to fingerprint research system. Because of a massive quantity of simultaneous requests are expected in cloud backup services.

In order to resolve this trouble, the hash cluster is designed for high load-balancing, scalability and minimizing the value for every fingerprint research question. The hash cluster is designed as middleware between the customers and the cloud storage. It offers the fingerprint storage and lookup carrier. There are other works on deduplication storages which their architectures are designed for scalability issue, for example; Extreme Binning [18], and Droplet [19].

Extreme Binning is used to construct a allotted record backup system. The architecture of such machine is composed of several backup nodes. Each backup node consists of a compute center and RAM in conjunction with a dedicated attached disk. The first undertaking while a record arrives to the system for backup is, it must be chunked. The system can delegate this mission to someone of the backup nodes via choosing one consistent with the system load at that point. After chunking, stateless routing set of rules is used to course the chunked report with the aid of the usage of its bite ID. The chunked document can be routed to a backup node where it is going to be deduplicated and stored.

Droplet, a distributed deduplication storage machine designed for excessive throughput and scalability. It includes 3 additives: a single meta server that monitors the entire machine reputation, a couple of fingerprinting servers that run deduplication on enter facts flow, and more than one storage nodes that keep fingerprint index and deduplicated facts blocks.

Meta server maintains statistics of fingerprinting and storage servers in the machine. When new nodes are added into the system, they want to be registered on the meta server first. The meta server provides a routing provider with this data. The consumer first connects to the meta server and queries for listing of fingerprinting servers, after which connects to one of them. After this, a uncooked statistics circulate containing backup content material can be sent to this fingerprinting server, which calculates facts block fingerprints and replies consequences to the patron. Fingerprint servers check duplicated fingerprint through querying storage servers.

The nature of facts in cloud storage dynamic [20], [21]. For instance, facts usage in cloud changes time beyond regulation, a few records chunks may be study often in time period, however won't be used in all over again period. Some datasets may be frequently accessed or updated with the aid of multiple users on the identical time, at the same time as others may additionally want the high level of redundancy for reliability requirement. Therefore, it is essential to assist this dynamic characteristic in cloud storage. However, modern-day approaches are ordinarily targeted on static scheme, which limits their full applicability in dynamic characteristic of records in cloud storage.

III. PROPOSED SYSTEM MODEL

A. Overall Architecture

Our machine is presently primarily based on client-side deduplication using entire record hashing. Hashing method is carried out on the consumer, and connects to any individual of Deduplicators in line with their hundreds at that time. The deduplicator then identifies the duplication by evaluating with the prevailing hash values in Metadata Server. In traditional deduplication structures, if it is a new hash price, it is going to be recorded in metadata server, and the record will be uploaded to File Servers, its logical path may even recorded in metadata server. If it does exist, the number of references for the report will be accelerated. Some structures may also preserve a number of copies of each record with a static range. However, the documents with a huge wide variety of references can also require more replicas for you to enhance availability. To resolve this trouble, a few existing works introduced level of redundancy into deduplication structures.

However, figuring out stage of redundancy with the aid of quantity of references is a negative measurement because files with fewer references may be crucial documents. In order to improve availability at the same time as maintaining storage performance, we suggest a deduplication system which considers each the dynamicity and taking Quality of Service (QoS) of the Cloud surroundings into attention. In our machine model, after identifying the duplication, the Redundancy Manager then calculates an choicest variety of copies for the file based totally on range of references and stage of QoS vital. The numbers of copies are dynamically changed based at the changing range of references, stage of QoS and call for for the files. The adjustments are monitored, for instance, while a document is deleted by means of a consumer, or the level of QoS of the file has been up to date, this will cause the redundancy manager to re-calculate most advantageous range of copies.

Our proposed device model is shown in figure 1. The device is composed of the subsequent components:

- Load Balancer: after hashing method with SHA-1, customers send a fingerprint (hash value) to a deduplicator through the burden balancer.

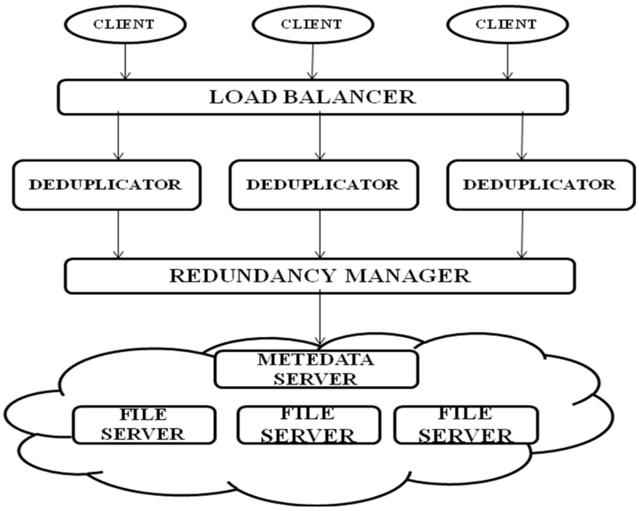


Figure : System model

The load balancer responds to requests from clients sending to anybody of deduplicators in step with their masses at that point.

- Deduplicators: a element designed for figuring out the duplication by means of evaluating with the existing hash values saved in metadata server.
- Cloud Storage: a Metadata Server to shop metadata, and some of File Servers to store actual files and their copies.
- Redundancy Manager: a factor to pick out the preliminary quantity of copies, and reveal the changing degree of QoS.

IV. SIMULATION ENVIRONMENT

HDFS Simulator is more relevant to our work, as HDFS Simulator already provides replication mechanisms, despite the fact that the replication degree is a predefined and static cost. However, it's far feasible to adjust the supply code if you want to introduce replication dynamicity. Moreover, we are able to perform experiments via simulating activities just like the converting degree of QoS. The mechanism of this painting is evaluated through the usage of simulation, because it enables researchers an opportunity to simulate huge-scale cloud environments, particularly failure occasions in the cloud as well as assist in evaluation QoS metrics which includes availability and overall performance.

The ideas of HDFS Simulation have been adapted to simulate our proposed device model. We create one Name node as Metadata server, and five Data nodes as File servers. Metadata in XML layout is saved in metadata server. File servers store the copies of documents. There are three occasions which we simulated: upload, replace, and delete. The upload event is while the report is first uploaded to the device. If documents already exist in the machine, and had been uploaded again, the quantity of copies of the files could be recalculated according to the best level of QoS, that is for an update occasion. For a delete record event, users can delete their documents, however the files will not completely deleted from the device if there are every other users confer with the identical documents.

A. Upload

Deduplicator calls a hash price of the uploaded document from client, after which exams for any duplicates with the same present hash price in metadata server. If it is a brand new document, the new metadata of the file might be introduced to the machine and the file might be uploaded to file server. The replicas of the file will be created in line with the level of QoS of the add report.

B. Update

In the case of current record, the metadata of the report might be updated and the device may also want to create or delete the replicas of the document consistent with the maximum fee of QoS of the record.

C. Delete

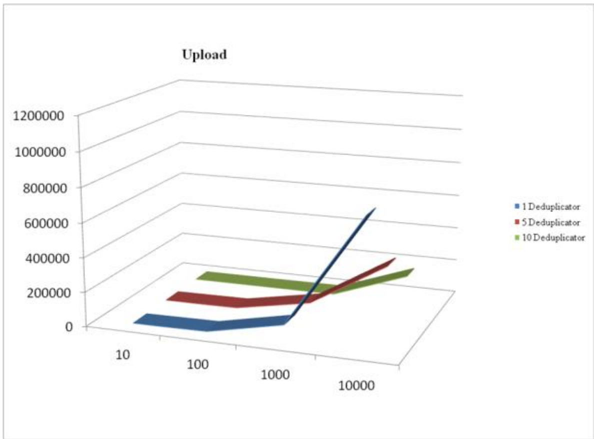
The deduplicator checks the quantity of documents which talk over with the same hash value person desires to delete. If there may be handiest one reference to the hash, all replicas of the report may be deleted. On the other hand if there are every other files that talk to the hash, handiest the metadata may be updated, and the quantity of replicas of the record might also need to lower consistent with the most fee of QoS.

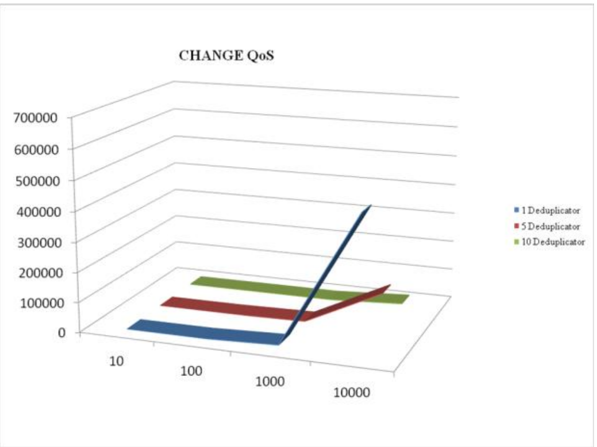
V. EXPERIMENTAL RESULTS

We perform experiments at the simulation of our proposed version. The experiments are completed for one, five, and ten deduplicators. all the documents used within the experiments have been created with stochastic contents and properties. For trying out the converting level of QoS, each file has been randomly assigned its degree of QoS (1-5). A single QoS value of 1-5 indicates the level of redundancy of every report. documents with higher stage of QoS will be replicated greater than the decrease ones. when a single deduplicator is used, the system faced scalability troubles taking an extended time when the range of documents extended as proven in figure 2.

This is due to the fact underneath the heavy load with greater requests and extra customers, a single deduplicator cannot maintain the overall performance of the machine. while the quantity of deduplicators is elevated to 5 and ten, the results show that it helps to reduce the processing time.

For uploads, when all the documents have been uploaded to the device for the primary time, comparing the time taken by using one deduplicator to five and ten deduplicators. including extra deduplicators when the number of upload documents increase, could help to lessen the processing time.





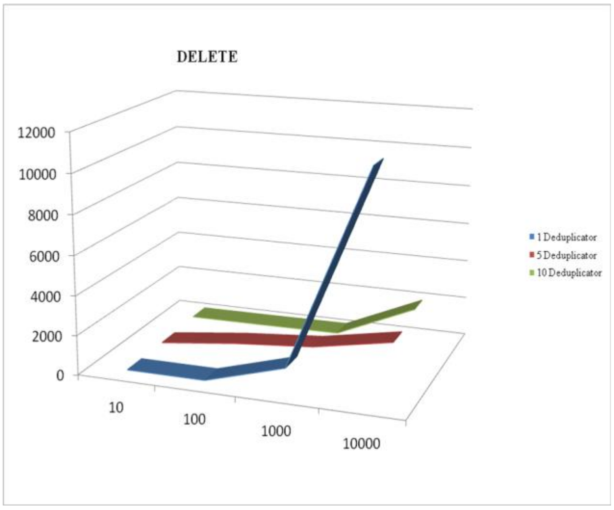


Figure 2: Experimental Results

whilst files have have already got been uploaded to the system, we carry out experiments for the case when there may be a changing level of QoS, because of this the number of copies of files in the device could be modified consistent with the most cost of QoS. The effects of replace documents display that once the number of documents growth, including more deduplicators can assist to lessen the processing time. We found that, whilst the numbers of files are ten, one hundred and one thousand, time saving via including more deduplicators are less than time saving for the add cases. but, whilst the numbers of files are extended to at least one thousand and ten thousands files, the time saving through 5 and ten deduplicators still boom, in comparison to the upload instances. We perform experiments to delete files. Including extra deduplicators can also to lessen the processing time, however the consequences of delete files are slightly different from the upload and replace cases. We are able to see that, for the delete case, times saving by using including extra deduplicators are decreased when the numbers of documents are expanded from ten to one hundred and one thousand files. but, while the numbers of documents are multiplied to 10 lots, extra deduplicators assist to increase time saving.

VI. CONCLUSION

Cloud storage offerings furnished in cloud computing has been increasing in reputation. It gives on call for virtualized storage property and clients simplest pay for the gap they sincerely consumed. As the growing call for and records keep inside the cloud, records deduplication is in all likelihood one of the techniques used to decorate storage efficiency. Nevertheless, modern information deduplication mechanisms in cloud storage are dynamic scheme, which video display units the QoS (nice of service) and NoF (number of failure). The proposed system uses a dynamic statistics deduplication scheme for cloud storage, in order to fulfill stability among converting storage efficiency and fault tolerance specifications, and also to enhance performance in cloud storage applications. We dynamically trade the wide variety of copies of files according to the changing stage of QoS, number of failure (NoF) and changing of clients' call for of files. The experimental effects show that, our proposed machine is performing well and can manipulate with scalability hassle.

REFERENCES

[1] I. Foster, Z. Yong, I. Raicu, and S. Lu, "Cloud Computing and Grid Computing 360-Degree Compared," in Grid Computing Environments Workshop, 2008. GCE '08, 2008, pp. 1-10.

[2] T. Dillon, W. Chen, and E. Chang, "Cloud Computing: Issues and Challenges," in Advanced Information Networking and Applications (AINA), 2010 24th IEEE International Conference on, 2010, pp. 27-33.

[3] T. G. Peter Mell, "The NIST Definition of Cloud Computing," National Institute of Standards and Technology NIST Special Publication 800-145, September 2011.

[4] SNIA Cloud Storage Initiative, "Implementing, Serving, and Using Cloud Storage," Whitepaper 2010.

[5] D. Harnik, B. Pinkas, and A. Shulman-Peleg, "Side Channels in Cloud Services: Deduplication in Cloud Storage," Security & Privacy, IEEE, vol. 8, pp. 40-47, 2010.

[6] S. Guo-Zi, D. Yu, C. Dan-Wei, and W. Jie, "Data Backup and Recovery Based on Data De-Duplication," in Artificial Intelligence and Computational Intelligence (AICI), 2010 International Conference on, 2010, pp. 379-382.

[7] SNIA, "Advanced Deduplication Concepts," 2011.

[8] V. Javaraiah, "Backup for cloud and disaster recovery for consumers and SMBs," in Advanced Networks and Telecommunication Systems (ANTS), 2011 IEEE 5th International Conference on, 2011, pp. 1-3.

[9] L. L. You, K. T. Pollack, and D. D. E. Long, "Deep Store: An Archival Storage System Architecture," presented at the Proceedings of the 21st International Conference on Data Engineering, 2005.

[10] T. Yujuan, J. Hong, F. Dan, T. Lei, Y. Zhichao, and Z. Guohui, "SAM:A Semantic-Aware Multi-tiered Source De-duplication Framework for Cloud Backup," in Parallel Processing (ICPP), 2010 39th International Conference on, 2010, pp. 614-623.

[11] S. Kumar Bose, S. Brock, R. Skeoch, N. Shaikh, and S. Rao, "Optimizing live migration of virtual machines across wide area networks using integrated replication and scheduling," in Systems Conference (SysCon), 2011 IEEE International, 2011, pp. 97-102.

[12] S. K. Bose, S. Brock, R. Skeoch, and S. Rao, "CloudSpider: Combining Replication with Scheduling for Optimizing Live Migration of Virtual Machines across Wide Area Networks," in Cluster, Cloud and Grid Computing (CCGrid), 2011 11th IEEE/ACM International Symposium on, 2011, pp. 13-22.

[13] N. Mandagere, P. Zhou, M. A. Smith, and S. Uttamchandani, "Demystifying data deduplication," presented at the Proceedings of the ACM/IFIP/USENIX Middleware '08 Conference Companion, Leuven, Belgium, 2008.

[14] D. Bhagwat, K. Pollack, D. D. E. Long, T. Schwarz, E. L. Miller, and J. F. Paris, "Providing High Reliability in a Minimum Redundancy Archival Storage System," in Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, 2006. MASCOTS 2006. 14th IEEE International Symposium on, 2006, pp. 413-421.

[15] F. Yinjin, J. Hong, X. Nong, T. Lei, and L. Fang, "AA-Dedupe: An Application-Aware Source Deduplication Approach for Cloud Backup Services in the Personal Computing Environment," in Cluster Computing (CLUSTER), 2011 IEEE International Conference on, 2011, pp. 112-120.

[16] T. Yujuan, J. Hong, F. Dan, T. Lei, and Y. Zhichao, "CABdedupe: A Causality-Based Deduplication Performance Booster for Cloud Backup Services," in Parallel & Distributed Processing Symposium (IPDPS), 2011 IEEE International, 2011, pp. 1266-1277.

[17] X. Lei, H. Jian, S. Mkandawire, and J. Hong, "SHHC: A Scalable Hybrid Hash Cluster for Cloud Backup Services in Data Centers," in Distributed Computing Systems Workshops (ICDCSW), 2011 31st International Conference on, 2011, pp. 61-65.

[18] D. Bhagwat, K. Eshghi, D. D. E. Long, and M. Lillibridge, "Extreme Binning: Scalable, parallel deduplication for chunk-based file backup," in Modeling, Analysis & Simulation of Computer and Telecommunication Systems, 2009. MASCOTS '09. IEEE International Symposium on, 2009, pp. 1-9.

[19] Z. Yang, W. Yongwei, and Y. Guangwen, "Droplet: A Distributed Solution of Data Deduplication," in Grid Computing (GRID), 2012 ACM/IEEE 13th International Conference on, 2012, pp. 114-121.

[20] W. Cong, W. Qian, R. Kui, C. Ning, and L. Wenjing, "Toward Secure and Dependable Storage Services in Cloud Computing," Services Computing, IEEE Transactions on, vol. 5, pp. 220-232, 2012.

[21] K. Yang and X. Jia, "An Efficient and Secure Dynamic Auditing Protocol for Data Storage in Cloud Computing," Parallel and Distributed Systems, IEEE Transactions on, vol. PP, pp. 1-1, 2012.

[22] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. De Rose, and R. Buyya, "CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," Software: Practice and Experience, vol. 41, pp. 23-50, 2011.

[23] C. Debains, P. A.-T. Togores, and F. Karakusoglu, "Reliability of Data- Intensive Distributed File System: A Simulation Approach," 2010.

[24] X. Jin, H. Yiming, L. Guojie, T. Rongfeng, and F. Zhihua, "Metadata Distribution and Consistency Techniques for Large-Scale Cluster File Systems," Parallel and Distributed Systems, IEEE Transactions on, vol. 22, pp. 803-816, 2011.

[25] O. Parisot, A. Schlechter, P. Bauler, and F. Feltz, "Flexible Integration of Eventually Consistent Distributed Storage with Strongly Consistent Databases," in Network Cloud Computing and Applications (NCCA), 2012 Second Symposium on, 2012, pp. 65-7